◆IEEE

IEEE Xplore
RELEASE 1.0

Membership Publications/Services Standards Conferences Careers/Jobs

Welcome
United States Patent and Trademark Office

» AB!

Help   FAQ   Terms   IEEE Peer Review | Quick Links ▼ |

Transient fault detection via simultaneous multithre

Reinhardt, S.K.   Mukherjee, S.S.
Dept. of Electr. Eng. & Comput. Sci., Michigan Univ., Ann Arbor, MI, USA;
This paper appears in: **Computer Architecture, 2000. Proceedings of the International Symposium on**

Abstract:
Smaller feature sizes, reduced voltage levels, higher transistor counts, and re
margins make future generations of microprocessors increasingly prone to tra
hardware faults. Most commercial fault-tolerant computers use fully replicatec
components to detect microprocessor faults. The components are locksteppec
cycle synchronized) to ensure that, in each cycle, they perform the same oper
the same inputs, producing the same outputs in the absence of faults. Unfortu
a given hardware budget, full replication reduces performance by statically pa
resources among redundant operations. We demonstrate that a Simultaneou
Redundantly Threaded (SRT) processor-derived from a a Simultaneous Mult
(SMT) processor-provides transient fault coverage with significantly higher p
An SRT processor provides transient fault coverage by running identical copi
same program simultaneously as independent threads. An SRT processor prov
performance because it dynamically schedules its hardware resources among
redundant copies. However, dynamic scheduling makes is difficult to impleme
lockstepping, because corresponding instructions from redundant threads ma'
execute in the same cycle or in the same order. This paper makes four contril
the design of SRT processors. First, we introduce the concept of the sphere of
which abstract both the physical redundancy of a lockstepped system and the
redundancy of an SRT processor. This framework aids in identifying the scope
coverage and the input and output values requiring special handling. Second,
two viable spheres of replication in an SRT processor, and show that one of th
provides fault detection while checking only committed stores and uncached l

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE *Xplore*
RELEASE 1.6

Welcome
United States Patent and Trademark Office

» ABS

Help   FAQ   Terms   IEEE Peer Review | Quick Links   ▾ |

Welcome to IEEE *Xplore*

○ Home
○ What Can
   I Access?
○ Log-out

Tables of Contents

○ Journals
   & Magazines
○ Conference
   Proceedings
○ Standards

Search

○ By Author
○ Basic
○ Advanced

Member Services

○ Join IEEE
○ Establish IEEE
   Web Account

○ Access the
   IEEE Member
   Digital Library

Search Results   [PDF FULL-TEXT 76 KB]   DOWNLOAD CITATION

Request Permissions
RIGHTSLINK◊
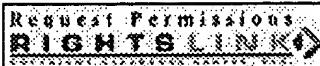
AR-SMT: a microarchitectural approach to fault tole in microprocessors

Rotenberg, E.
Dept. of Comput. Sci., Wisconsin Univ., Madison, WI, USA;
This paper appears in: **Fault-Tolerant Computing, 1999. Digest of Paper: Ninth Annual International Symposium on**

Meeting Date: 06/15/1999 - 06/18/1999
Publication Date: 15-18 June 1999
Location: Madison, WI USA
On page(s): 84 - 91
Reference Cited: 24
Number of Pages: xx+357
Inspec Accession Number: 6314443

Abstract:
This paper speculates that technology trends pose new challenges for fault tol microprocessors. Specifically, severely reduced design tolerances implied by c clock rates may result in frequent and arbitrary transient faults. We suggest t fault-tolerant techniques-system-level, gate-level, or component-specific appi either too costly for general purpose computing, overly intrusive to the desigr insufficient for covering arbitrary logic faults. An approach in which the microa itself provides fault tolerance is required. We propose a new time redundancy tolerant approach in which a program is duplicated and the two redundant pro simultaneously run on the processor: The technique exploits several significar microarchitectural trends to provide broad coverage of transient faults and re: coverage of permanent faults. These trends are simultaneous multithreading, and data flow prediction, and hierarchical processors-all of which are intendec performance, but which can be easily leveraged for the specified fault toleran The overhead for achieving fault tolerance is low, both in terms of performanc changes to the existing microarchitecture. Detailed simulations of five of the S benchmarks show that executing two redundant programs on the fault-tolerai microarchitecture takes only 10% to 30% longer than running a single versioi program

Index Terms:
computer architecture   fault tolerant computing   redundancy   AR-SMT   fault tolerance

Search Results   [PDF FULL-TEXT 1136 KB]   DOWNLOAD CITATION

Request Permissions
RIGHTSLINK

Simultaneous multithreading: Maximizing on-chip parallelism

Tullsen, D.M.   Eggers, S.J.   Levy, H.M.
Dept. of Comput. Sci. & Eng., Washington Univ., Seattle, WA, USA;
This paper appears in: **Computer Architecture, 1995. Proceedings. 22nd International Symposium on**

Abstract:
This paper examines simultaneous multithreading, a technique permitting
independent threads to issue instructions to a superscalar's multiple functiona
single cycle. We present several models of simultaneous multithreading ar
them with alternative organizations: a wide superscalar, a fine-grain multithr
processor, and single-chip, multiple-issue multiprocessing architectures. Our r
that both (single-threaded) superscalar and fine-grain multithreaded archite
limited in their ability to utilize the resources of a wide-issue processor. Simu
multithreading has the potential to achieve 4 times the throughput of a sup
and double that of fine-grain multi-threading. We evaluate several cache conf
made possible by this type of organization and evaluate tradeoffs between the
show that simultaneous multithreading is an attractive alternative to singl
multiprocessors; simultaneous multithreaded processors with a variety of
organizations outperform corresponding conventional multiprocessors with sin
execution resources. While simultaneous multithreading has excellent pote
increase processor utilization, it can add substantial complexity to the design.
examine many of these complexities and evaluate alternative organizations in
space

Index Terms:
cache configurations   computational complexity   computer architecture   fine-grain mult
processor   multiprocessing architectures   multiprocessing systems   on-chip parallelism
maximisation   processor scheduling   processor utilization   simultaneous multithread